

## Convertible energy meter.

**Publication number:** DE69228850T

**Publication date:** 1999-10-07

**Inventor:** ATHERTON KENNETH WILLIAM (US); GROGAN KEVIN PAUL (US); DASTOUS SUSAN DORIS (US); LAVOIE GREGORY PAUL (US); MANCUSO MARJORIE JO (US); PLIS MARK JOSEPH (US)

**Applicant:** GEN ELECTRIC (US)

**Classification:**

- international: **G01R21/133; G01R22/00; G01R21/00; G01R22/00;**  
(IPC1-7): G01R21/133; G01R11/58

- european: G01R21/133C

**Application number:** DE19926028850T 19920124

**Priority number(s):** US19910764863 19910924

Also published as:

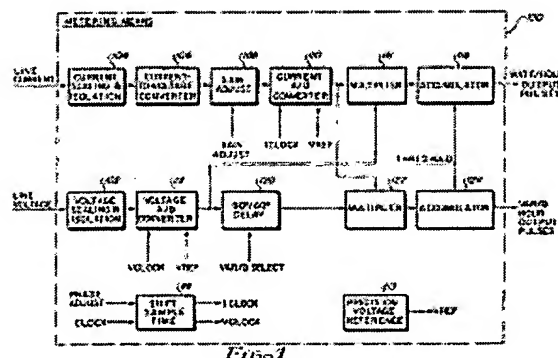
EP0534583 (A2)  
JP5119081 (A)  
EP0534583 (A3)  
EP0534583 (B1)

[Report a data error here](#)

Abstract not available for DE69228850T

Abstract of corresponding document: **EP0534583**

The present invention, in one embodiment, comprises an energy meter capable of converting from one mode of operation to another mode of operation. In the one embodiment, the present invention comprises means for measuring energy consumption, and means for converting operation of the measuring means from a first mode to a second mode.



Data supplied from the **esp@cenet** database - Worldwide

**THIS PAGE BLANK (USPTO)**



72

19 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

12 Übersetzung der  
europäischen Patentschrift

97 EP 0 534 583 B 1

10 DE 692 28 850 T 2

51 Int. Cl.<sup>6</sup>:  
G 01 R 21/133  
G 01 R 11/58

- 21 Deutsches Aktenzeichen: 692 28 850.3
- 95 Europäisches Aktenzeichen: 92 300 613.4
- 96 Europäischer Anmeldetag: 24. 1. 92
- 97 Erstveröffentlichung durch das EPA: 31. 3. 93
- 97 Veröffentlichungstag der Patenterteilung beim EPA: 7. 4. 99
- 47 Veröffentlichungstag im Patentblatt: 7. 10. 99

- 30 Unionspriorität:  
764863 24. 09. 91 US
- 73 Patentinhaber:  
General Electric Co., Schenectady, N.Y., US
- 74 Vertreter:  
Voigt, R., Dipl.-Ing., Pat.-Anw., 65812 Bad Soden
- 84 Benannte Vertragsstaaten:  
CH, DE, FR, GB, LI

72 Erfinder:  
Atherton, Kenneth William, Saco, Maine 04072, US;  
Grogan, Kevin Paul, South Berwick, Maine 03908,  
US; Dastous, Susan Doris, Milford, New Hampshire  
03055, US; Lavoie, Gregory Paul, Newmarket, New  
Hampshire 03857, US; Mancuso, Marjorie Jo,  
Exeter, New Hampshire 03833, US; Plis, Mark  
Joseph, Barrington, New Hampshire, US

54 Umwandelbarer Energiezähler

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patent- und Markenamt inhaltlich nicht geprüft.

DE 692 28 850 T 2

DE 692 28 850 T 2



Ein Teil der Beschreibung in diesem Patentdokument enthält Material, das dem Copyright- bzw. Urheberrechtsschutz unter-  
5 liegt. Der Inhaber des Copyrights hat keine Einwände gegen eine von einem Dritten vorgenommene Faksimile-Wiedergabe des Patentdokuments oder der Patentbeschreibung, wie sie in der Akte oder in den Aufzeichnungen des Patent- und Markenamtes erscheint, behält sich aber im übrigen alle irgendwie gearteten Urheber-  
10 rechte vor.

## I. Hintergrund der Erfindung

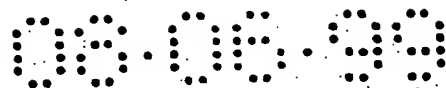
### A. Das Gebiet der Erfindung

15 Die vorliegende Erfindung bezieht sich im allgemeinen auf Energiemeßgeräte und betrifft in speziellerer Hinsicht Energiemeßgeräte mit Anpassungsfähigkeit zum Durchführen verschiedener Arten von Meßfunktionen.

### B. Verwandter Stand der Technik

Auf dem Gebiet der Energiemessung betrachten Fachleute auf diesem Feld typischerweise ein Meßgerät bzw. einen Zähler als aus zwei (2) grundlegenden Bauteilen zusammengesetzt - eine messende Komponente und eine Registrier- bzw. Registerkomponente.  
25 te. Die messende Komponente ist mit einem Netzwerk für die Energieverteilung verbunden und erzeugt auf den Leistungsverbrauch hinweisende Signale. Diese Signale werden an die Registerkomponente weitergeleitet, die derartige Funktionen wie das Speichern der Signale in einem Verwendungszeit- oder in einem  
30 Bedarfstyp-Format durchführt.

Etwa einmal im Monat wird ein Zählerableser auf den Weg geschickt, um den Zähler "abzulesen". In einigen Fällen betrifft das Ablesen eines Zählers schlicht das Herunterladen des Spei-  
35 chers der Registerkomponente in ein von dem Zählerableser mitgeführtes tragbares elektronisches Speichergerät.



In jüngerer Zeit haben Meßgerätehersteller damit begonnen, die messende Komponente sowie die Registerkomponente vollständig als Festkörperschaltung herzustellen. Dieser Übergang beruht zumindest zum Teil auf dem Wunsch der Energieversorgungsunternehmen nach Energiemeßgeräten bzw.-zählern mit der Flexibilität für die Durchführung eines breiten Bereichs von Funktionen. Eine solche Flexibilität kann nur unter Einsatz der Festkörpertechnologie bereitgestellt werden. Ein Beispiel eines derartigen Meßgeräts für elektrische Energie, das eine Festkörper-Meßeinheit benutzt, ist in der GB-A-1,575,148 beschrieben.

Eine mit dem Übergang auf die Festkörpertechnologie mögliche bedeutsame und potentiell Kosten einsparende Eigenschaft besteht in der Schaffung von "umwandelbaren" Zählern. Dies bedeutet, daß die Registerkomponente geändert werden kann, zum Beispiel von der Durchführung von Nur-Bedarfs-Funktionen auf Bedarfs-und Verwendungszeit-Funktionen. Die Registerkomponente ist damit wandelbar von einer Funktion zu anderen Funktionen.

Bis jetzt konnte man jedoch eine solche Umwandelbarkeit nur erreichen, indem man die Registerkomponente herausnahm und dafür eine andere Registerkomponente an deren Stelle setzte, die die gewünschte Funktion zu leisten in der Lage war. Es würde zum Beispiel eine Nur-Bedarfs-Registerkomponente aus dem Zähler bzw. Meßgerät entnommen und durch eine Bedarfs-und Verwendungszeit-Registerkomponente ersetzt werden. Anstatt "umwandelbar" zu sein, würden diese Registerkomponenten tatsächlich "austauschbare" Module darstellen. Ein Beispiel einer modularen Registerkomponente ist in dem US Patent No. 5,014,213 ausgeführt.

Es ist deshalb wünschenswert, ein in Wahrheit umwandelbares Register vorzusehen, das kein Entfernen und Ersetzen einer Registerkomponente gegen eine andere erfordert, um die Registerfunktion zu ändern.

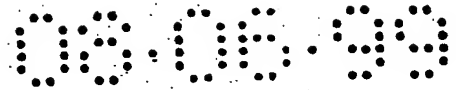
## II. Zusammenfassung der Erfindung

Die vorliegende Erfindung schafft eine programmierbare elektronische Festkörper-Registerkomponente für ein elektrisches  
5 Energiemeßgerät, wie sie im Anspruch 1 der beigefügten Ansprüche beansprucht ist. In einer Ausführungsform wird ein Betriebssystem in den Anwendungen der Registerkomponenten benutzt. Das Betriebssystem koordiniert die Abfolge der Aufgabendurchführung sowie die Prioritäten der von der Registerkomponente auszuführenden hauptsächlichen Aufgaben. Das Betriebssystem  
10 wird in einer Ausführungsform über Tabellen betrieben. Die Tabellen stellen eine Form einer indirekten Adressierung für entsprechende Software-Routinen dar, die den Betrieb der Registerkomponente für die Durchführung entsprechender Aufgaben in  
15 einer vorbestimmten Reihenfolge steuern.

Die Tabellen werden vom RAM aus ausgeführt und können auf Software-Routinen zeigen, die im ROM oder RAM gespeichert sind. Im Einzelnen sind die Aufgaben-Tabellen zusammengesetzt aus  
20 Zeigern für Module, die entsprechende Aufgaben ausführen, zum Beispiel die Zeiteinrichtungen bzw. Timer und Zähler für Informationen zu "Zeit und Datum" im Auge behalten. Durch Änderung des Betriebsmodus des Betriebssystems wird ein unterschiedlicher Satz von Aufgaben ausgeführt.

25

Im normalen Ausführungsmodus vollzieht das Operationssystem eine Endlosschleife, d.h. solange keine Markierung (flag) für eine Unterbrechung oder eine "Neue Priorität" gesetzt wird, fährt das Betriebssystem in der Ausführung der Schleife fort.  
30 Diese Schleife wird hier manchmal als die "Kernel"-Schleife angesprochen. Am Beginn der Kernel-Schleife wird ein MODE Byte eingelesen, um einen Eintrag in einer Mode-Tabelle (MODE TABLE) zu indexieren. Jeder der Einträge in der MODE TABLE stellt einen Zeiger zu einem Eintrag in der Aufgaben-Tabelle dar.  
35 Jeder Eintrag in der Aufgaben-Tabelle ist ein Zeiger zu einer Software-Routine.



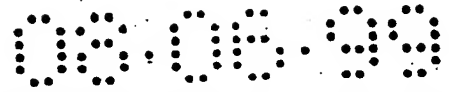
Das Operationssystem veranlaßt, daß die erste Aufgabe aus der ausgewählten Aufgaben-Tabelle ausgeführt wird, und prüft dann, ob eine Anforderung für einen unmittelbaren Moduswechsel vorliegt. Solange keine Anforderung für einen unmittelbaren Moduswechsel vorliegt, werden die Aufgaben in der in der ausgewählten Aufgaben-Tabelle definierten Folge ausgeführt, und zwar bis das Ende der ausgewählten Aufgaben-Tabelle erreicht ist, d.h. bis die letzte Aufgabe in der ausgewählten Aufgaben-Tabelle ausgeführt ist. Wenn das Ende der ausgewählten Aufgaben-Tabelle erreicht ist, oder wenn ein unmittelbarer Moduswechsel angefordert wird, lädt das Operationssystem erneut den Wert des MODE Bytes und indexiert die MODE TABLE erneut. Es wird sodann die neu ausgewählte Aufgaben-Tabelle ausgeführt.

Wenn kein Moduswechsel, weder ein unmittelbarer noch ein regulärer, angefordert worden ist, wird die "aufgezeigte" Aufgaben-Tabelle dieselbe Aufgaben-Tabelle sein, die gerade ausgeführt wird. Als wichtiger Punkt ist festzuhalten, daß, wenn ein Moduswechsel auftritt, die Aufgabe 0 in der ausgewählten Modus-Tabelle die erste Aufgabe ist, die ausgeführt wird. Wenn kein Moduswechsel aufgetreten ist, und die gerade ausgeführte Aufgaben-Tabelle erneut ausgewählt wird, wird die Aufgabe 1 der Aufgaben-Tabelle als erste Aufgabe ausgeführt.

Die Aufgabe 0 von jeder Aufgaben-Tabelle wird als die Mode-Initialisierungsaufgabe bezeichnet und stellt eine Stelle bereit, wo Routinen angeordnet sein können, die lediglich einmal pro Modus ausgeführt zu werden brauchen, anstelle einmal pro Aufgaben-Tabellenzyklus im Verlauf des Modus. Wenn keine Routinen in einem Modus vorliegen, die dieses Erfordernis haben, gibt die Mode-Initialisierungsaufgabe die Steuerung schlicht an das Betriebssystem zurück, welches dann die Ausführung der Aufgaben-Tabelle, beginnend mit der Aufgabe 1, weiterführt.

35

Um in der einen Ausführungsform zum Beispiel eine Registerkomponente mit dem gegenwärtigen Betriebssystem umzuwandeln von



- einem Nur-Bedarfs-Register in ein Verwendungszeit- und Bedarfs-Register, werden die Aufgaben-Tabellen in die Registerkomponente geladen, und es wird ein anderer Modus ausgewählt. Die zusätzlichen Aufgaben-Tabellen werden für die Durchführung
- 5 notwendiger Verwendungszeit-Funktionen benutzt. Indem man einen geeigneten Modus auswählt, wird die Ausführungssequenz der Aufgaben-Tabelle derart geändert, daß die passenden Aufgaben-Tabellen abgearbeitet werden.
- 10 Die vorliegende Erfindung sieht vor, und das ist von Bedeutung, daß eine Registerkomponente geändert werden kann von einem Nur-Bedarfs-Register in ein Verwendungszeit- und Bedarfs-Register, und zwar ohne daß man die Registerkomponente entnehmen und ersetzen muß. Die vorliegende Erfindung erleichtert die Schaf-
- 15 fung eines in Wahrheit umwandelbaren Registers.

### III. Kurze Beschreibung der Zeichnungen

- Diese sowie weitere Aufgaben der vorliegenden Erfindung werden
- 20 zusammen mit weiteren Merkmalen und Vorteilen davon aus der folgenden detaillierten Beschreibung deutlich werden, wenn man sie zusammen mit den beigegeführten Zeichnungen liest. Darin zeigen:

- 25       Figur 1 eine blockschaltbildmäßige Beschreibung einer Ausführungsform für die Meßeinrichtung, wie sie mit der vorliegenden Erfindung benutzt werden kann;
- Figur 2 eine blockschaltbildmäßige Beschreibung einer Ausführungsform einer Registriereinrichtung;
- 30       Figur 3 eine Darstellung in Form eines Flußdiagramms für eine Ausführungsform einer Abfolge von Prozeßschritten gemäß der vorliegenden Erfindung; und
- Figur 4 stellt eine Modus-Tabelle sowie Aufgaben-Tabellen dar.





#### IV. Detaillierte Beschreibung der Zeichnungen

Figur 1 ist eine blockschaltbildmäßige Beschreibung für eine Ausführungsform der Meß- bzw. Zählereinrichtung 100, die mit der vorliegenden Erfindung benutzt werden kann. Im Einzelnen sind Netzstrom- und Netzspannungseingänge für die Meßeinrichtung 100 vorgesehen, und es werden von der Meßeinrichtung 100 Ausgangsimpulse für die Wattstunden sowie für die VAR/Q-Stunden ausgegeben. Die Wattstunden-Ausgangsimpulse bzw. die Impulse für die VAR/Q-Stunden sind proportional zum Verbrauch an Wirk- bzw. Blindleistung.

Die Netzspannungs- und Netzstromeingänge sind elektrisch isoliert und präzise skaliert über entsprechende Skalier- und Isolationseinrichtungen 102 und 104, um sekundäre Signale zu liefern, die mit elektronischen Schaltkreisen kompatibel sind. Eine Spannungsskalierung zur Bereitstellung von  $1,6V_{eff}$  bei einer Nenn-Eingangsspannung (zum Beispiel 120 V) ist dazu geeignet. Stromskalierverhältnisse von 100.000 zu eins bzw. von 10.000 zu eins können für in sich abgeschlossene Meßgeräte (200A wahre Größe) bzw. für mit einem Übersetzungstransformator ausgelegte Meßgeräte (20A wahre Größe) gewählt werden. Die wahre Größe des Sekundärstroms von dem Stromfühler beträgt dann  $2,0 mA_{eff}$ . Ein hochgenauer Strom/Spannungsumsetzer 106 in der Stromeingangsleitung konvertiert ein Stromsignal in eine Spannung für Zwecke der Kompatibilität mit einem A/D (Analog-zu-Digital) Umsetzer. Die Skalierung des Strom/Spannungsumsetzers beträgt 1 Volt am Ausgang pro mA am Eingang (1.000 Ohm).

Es sind Einstellungen für die Verstärkung vorgesehen, indem man eine Verstärkerstufe 108 in dem Strompfad einstellt. Ein erster A/D Umsetzer 110 ist für das Stromsignal vorgesehen, und ein zweiter A/D Umsetzer 112 ist für das Spannungssignal vorgesehen. Beide A/D Umsetzer weisen einen Vollausschlagsbereich von ungefähr  $\pm 3,45$  Volt Gleichspannung auf, wie er durch VREF von einer Präzisionsreferenzspannung 113 bestimmt wird. Die genaue Zeitbasis (TAKT) 114 bestimmt eine konstante Abtastrate, mit der die A/D Konverter gemeinsam die Strom- bzw. die Span-

- nungseingänge "abtasten" und deren Amplituden in Binärwörter umwandeln. Es werden Abtastraten über einige Kilohertz hinaus erfordert, um eine gute Leistungsfähigkeit für Harmonische in den Eingangssignalen zu erhalten. Eine Phaseneinstellung, um
- 5 präzise die Phase der Strom- und Spannungseingangssignale abzugleichen, wird vorgesehen, indem man die Abtastzeit des Spannungsumsetzers relativ zu dem Stromumsetzer in kleinen diskreten Schritten verschiebt.
- 10 Zum Erhalt von Ausgangsimpulsen, die proportional zu den Wattstunden sind, wird jede binär kodierte Stromabtastung in einem Multiplizierer 116 mit seinem entsprechenden Spannungsabtastwert bzw. Sempel multipliziert, und das Produkt wird auf einen Akkumulator 118 hinzuaddiert. Jedesmal, wenn die akkumulierte
- 15 Summe einen Schwellenwert erreicht, der proportional ist zu der Wattstundenkonstante des Zählers, wird ein Ausgangsimpuls erzeugt. Die Ausgangsimpulsrate wurde so gewählt, daß sie zwölfmal die Rate einer Scheibenumdrehung für einen äquivalenten elektromechanischen Zähler ist, um Kompatibilität mit
- 20 früheren Generationen von Meßeinrichtungen aufrechtzuerhalten.
- Ausgangsimpulse, die proportional zu VAR-Stunden oder Q-Stunden sind, werden auf dieselbe Weise wie für die Wattstunden erhalten mit der Ausnahme, daß die benutzten Spannungsabtastwerte
- 25 über eine Verzögerungseinheit 120 verzögert werden, und zwar um eine Zeit, die äquivalent zu 90 Grad ist, für VAR-Stunden, oder um eine Zeit, die äquivalent zu 60 Grad ist, für Q-Stunden, wobei jede von beiden gewählt werden kann. Jedes binär kodierte Stromsempel wird in einem Multiplizierer 122 mit dem entsprechenden Spannungssempel multipliziert, und ein separater Akkumulator 124 wird für die Akkumulierung der VAR- oder Q-Stunden benutzt. Für die Schwellenwerte der VAR- oder Q-Stunden wird dieselbe Schwelle benutzt wie für die Schwelle des Wattstunden-Akkumulators. Ein typischer Schwellenwert beträgt  $144 (10^{-6})$
- 30 Volt-Ampere-Sekunden (für einen in sich abgeschlossenen Element-Zähler für Zweidraht-Einphasenanwendungen).
- 35

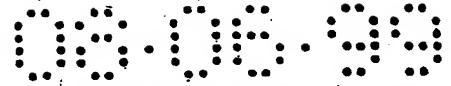


Für Mehrphasenbelastungen kann ein (nicht gezeigter) Multiplexer benutzt werden, um die Zählerfunktion zu erweitern. Es sollte eine separate Isolation und Skalierung für jeden Strom- und Spannungseingang vorgesehen werden, wobei die übrigen  
5 Elemente jedoch "zeitlich aufgeteilt" werden können.

Die Genauigkeit wird in erster Linie begrenzt durch Rauschen, die Auflösung des A/D Umsetzers, die Linearität der Eingangsskalierung, die Linearität des Strom/Spannungsumsetzers sowie  
10 durch die Linearität der A/D Umsetzer. Die Stabilität des Betriebs hinsichtlich der Zeit und Temperatur wird begrenzt durch die Stabilität der Eingangsskalierung, die Stabilität des Widerstandes im Strom/Spannungsumsetzer sowie durch die Stabilität der Zeitbasis und der Referenzspannung. Eine augenschein-  
15 liche Ungenauigkeit kann bei hohen Ausgangsimpulsraten in Kombination mit einigen wenigen Ausgangsimpulsen pro Messung auftreten. Diese Situation tritt auf wegen der zyklischen Natur des Leistungsflusses bei der doppelten Netzfrequenz, was ein "Zittern" auf den Ausgangsimpulsen verursacht. Dies läßt sich  
20 überwinden, indem man die Anzahl der Ausgangsimpulse pro Messung erhöht.

Figur 2 stellt eine blockschaltbildmäßige Beschreibung der Registerkomponenteneinrichtung 150 dar. Die im Blockschaltbild  
25 enthaltenen Funktionen können zum Beispiel bereitgestellt werden auf einer 1,2 Mikrometer CMOS anwendungsspezifischen integrierten Schaltung (ASIC von Application Specific Integrated Circuit), wie das auf dem Fachgebiet wohl bekannt ist.

30 Die ASIC enthält eine Mikroprozessorsteuerung 152, die mit einem Adress-, Daten-, Steuerungs- und Umacro-Bus 154 (nachfolgend als der "Bus" bezeichnet) verbunden ist. Eine Steuerung 156 für eine LCD- bzw. Flüssigkristallanzeige ist direkt mit dem Bus 154 sowie indirekt über eine logische  
35 Schnittstelle 158 mit dem Bus 154 gekoppelt. Ein mit einem Oszillator 162 gekoppelter Timer 160 für Stromausfall ist mit dem Bus 154 verbunden. Ein Speicher 164, der zum Beispiel



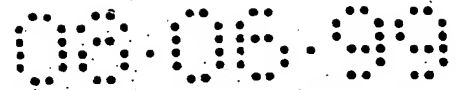
Speicherblöcke vom Typ ROM, EEPROM und SRAM enthält, ist ebenfalls an den Bus 154 angeschlossen. Die ASIC 150 enthält weiterhin (aber nicht gezeigt) eine Adressdekodierlogik für die Bestimmung der ASIC Speicherkarte, eine Gruppenschaltungslogik für die Erweiterung des Adressraums des Prozessors sowie eine Verbindungsfreigabelogik, um die Registerdaten für Programmierungs- und Wiedergabezwecke zu leiten.

Im Betrieb führt die Mikroprozessorsteuerung 152 Berechnungs- sowie ASIC Steuerungsfunktionen aus. Die Flüssigkristall-Display-Steuerung 156 wird verwendet, um Eingabe/Ausgabeeinrichtungen zu steuern, zum Beispiel eine Flüssigkristallanzeige. Der mit dem Oszillator 162 gekoppelte Timer 160 für Netzausfall wird für die Zeiterfassung im Falle eines Stromausfalls benutzt. Die Meßdaten, Programme und andere Informationen sind in dem Speicher 164 gespeichert. Weitere Details im Hinblick auf Energieverbrauchsmeasurements finden sich in EP-A-0,510,956, die ein Verfahren zum Betrieb eines Zählers zur Durchführung der Schritte angibt: Erzeugen eines Meßwerts für die verbrauchte Wirkenergie, Erzeugen eines Meßwerts für die verbrauchte Blindenergie und, aus den Meßdaten der Wirk- und Blindenergie, Erzeugen eines Meßwerts für die verbrauchte Scheinenergie.

Figur 3 ist ein Flußdiagramm 200, das eine Ausführungsform für eine Abfolge von Prozeßschritten gemäß der vorliegenden Erfindung zeigt. Im Einzelnen wird ein Aspekt der Erfindung bezeichnet als ein Betriebssystemkernel (os\_kernel). Das Betriebssystem steuert die Betriebsweise der zuvor beschriebenen Hardwarekonfiguration, und der Begriff "Kernel" bezieht sich auf den Teil des Programms, der die Abfolge durch die jeweiligen Aufgaben-Tabellen steuert. Das Flußdiagramm 200 stellt den Algorithmus für die Ausführung entsprechender Aufgaben-Tabellen (task tables) dar.

35

Im Einzelnen gilt, daß, wenn das os\_kernel initiiert ist, wie im Startblock 202 angezeigt, der nächste Schritt 204 darin



- besteht, den `task_table_index` auf Null zu setzen. Wie im Schritt 206 angezeigt, besteht die nächste Operation sodann darin, die `task_table` gleich der Tabelle unter der Adresse zu setzen, die in der mit dem `current_os_mode` indexierten `mode_table` angegeben ist. Im Einzelnen wird das Mode-Byte ausgelesen, um einen Eintrag in der Mode-Tabelle zu indexieren. Jeder der Einträge in der Mode-Tabelle ist ein Zeiger für eine Aufgaben-Tabelle. Unter Bezugnahme auf Figur 4 ist zum Beispiel in der Mode-Tabelle 300 der erste Eintrag ein Zeiger auf die Aufgaben-Tabelle 302. Ein nachfolgender Eintrag ist ein Zeiger auf die Aufgaben-Tabelle 304. Jeder Eintrag in der Aufgaben-Tabelle stellt einen Zeiger auf eine Haupt-Subroutine für eine High-Level Aufgabe dar.
- 15 Für die erste Iteration führt sodann das `os_kernel` die erste Aufgabe aus der Aufgaben-Tabelle aus. Es wird dann entschieden, ob das Ende der `task_table` erreicht worden ist, wie beim Schritt 208 angedeutet. Dies bedeutet, daß festgestellt wird, ob die letzte auszuführende Aufgabe in einer Aufgaben-Tabelle ausgeführt worden ist. Wenn die letzte Aufgabe ausgeführt worden ist, geht der Ablauf weiter zum Schritt 210, in dem der `previous_os_mode` auf den `current_os_mode` gesetzt wird, und der `current_os_mode` auf den `next_os_mode` gesetzt wird, wie im Schritt 212 angezeigt ist. Dies bedeutet, daß zumindest im Hinblick auf die Schritte 210 und 212 der Betriebsmodus nicht geändert worden ist, und daß dieselben Aufgaben-Tabellen auszuführen sind.

- Sodann wird im Schritt 214 bestimmt, ob ein Moduswechsel vorgenommen worden ist. Wenn ein Moduswechsel vorgenommen wurde, wird der `task_table_index` auf Null gesetzt, so daß die anfängliche Aufgabe in der auszuführenden Aufgaben-Tabelle als erste durchgeführt wird. Ist jedoch kein Moduswechsel erfolgt, dann wird der `task_table_index` gleich der `FIRST_REPEATED_TASK` gesetzt. Dies bedeutet, daß, wenn eine Aufgaben-Tabelle neu zum ersten Mal auszuführen ist, die den Modus initialisierende Aufgabe (Index = 0) durchzuführen ist (Schritt 216). Wurde der

Modus jedoch nicht geändert und ist die Aufgaben-Tabelle erneut auszuführen, muß die den Modus initialisierende Aufgabe nicht nochmal ausgeführt werden (Schritt 218). Das heißt, die den Modus initialisierenden Aufgaben sind lediglich einmal pro

5 Modus durchzuführen anstatt einmal pro Aufgaben-Tabellenzyklus während des Modus. Sobald der Aufgaben-Tabellenindex auf die richtige Einstellung gesetzt ist, kehrt der Betriebsablauf zum Schritt 206 zurück.

10 Wenn das Ende der task\_table nicht erreicht worden ist, wie beim Schritt 208 angezeigt, geht der Ablauf weiter zum Schritt 220. Im Schritt 220 wird festgestellt, ob ein sofortiger Moduswechsel angefordert worden ist. Wenn ein sofortiger Moduswechsel angefordert worden ist, geht der Ablauf weiter zum Schritt

15 210 und die hier zuvor beschriebenen Abläufe werden ausgeführt.

Wenn jedoch kein sofortiger Moduswechsel angefordert worden ist, wird die task\_page gleich der RAM page der in der task\_table spezifizierten Aufgabe gesetzt, d.h. die nächste

20 Aufgabe wird für ihre Durchführung ausgewählt. Die task\_address wird auf die Adresse der in der task\_table spezifizierten Aufgabe gesetzt, wie bei einem Schritt 224 angezeigt ist.

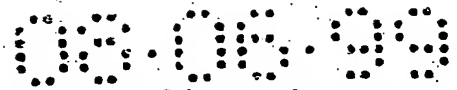
Wenn die bezeichnete task\_address sich in dem in das RAM heruntergeladenen Code befindet, wie beim Schritt 226 angezeigt ist, geht der Betriebsablauf weiter zum Schritt 228, wo die

25 code\_ram\_page gleich der task\_page gesetzt wird. Wenn die task\_address sich jedoch nicht in dem in das RAM heruntergeladenen Code befindet, wird die data\_ram\_page gleich der

30 task\_page gesetzt, wie im Schritt 230 angezeigt ist.

Der nächste Schritt 232 besteht dann darin, die Routine bei der task\_address aufzurufen und die nächste Aufgabe auszuführen. Der task\_table\_index wird sodann erhöht bzw. inkrementiert. Der

35 Ablauf kehrt dann zurück zum Schritt 208.



Die vorliegende Erfindung liegt in dem vorliegenden Algorithmus für ein Betriebssystem-Kernel für eine Verwendung in einem Elektrizitätszähler. Die besonderen Betriebsmoden und Aufgaben können von einem Systemoperator auf eine Weise definiert werden, die konsistent ist mit den Maßgaben, wie der Zähler zu  
5 nutzen ist. Es können zum Beispiel eine Vielzahl von Moden implementiert werden, zum Beispiel ein Verwendungszeit-Modus, ein Nur-Bedarfs-Modus, ein ausfallsicherer Modus, ein (nach einem Ausfall) Aufhol-Modus sowie ein Kommunikations-Modus.

10

Wenn, wie hier zuvor beschrieben, ein Modus ausgewählt wird, können Aufgaben-Tabellen aus dem ROM oder EEPROM in den RAM Speicherblock heruntergeladen werden. Die Abarbeitung der Aufgaben-Tabelle kann durchgeführt werden lediglich von dem RAM  
15 Speicherblock oder von einer Kombination von entsprechenden Speicherblöcken. Wenn ein Moduswechsel auftritt, müssen andere Aufgaben-Tabellen ausgeführt werden und können in den RAM heruntergeladen werden, um die Aufgaben-Tabellen von dem zuvor ausgewählten Modus zu ersetzen. Das hier zuvor beschriebene  
20 Operationssystem-Kernel ändert sich jedoch nicht, wenn ein anderer Modus ausgewählt wird.

Der Anhang bzw. Appendix A stellt eine Auflistung von Pseudocode für die Durchführung einer Ausführungsform des Operationssystemkernels dar. Der Anhang bzw. Appendix B ist eine Auflistung der Nur-Bedarfs-Modus-Tabelle. Appendix C ist eine Liste  
25 von Nur-Bedarfs-Aufgaben-Tabellen. Appendix D ist eine Auflistung der Verwendungszeit- und Bedarfs-Modus-Tabelle, und Appendix E ist eine Auflistung der Aufgaben-Tabellen für den  
30 Verwendungszeit-Betrieb. Diese Pseudokode-Listen werden bereitgestellt, um eine Ausführungsform eines von einer Aufgaben-Tabelle her gesteuerten Betriebssystems für einen Elektrizitätszähler anhand eines Beispiels näher zu erläutern.

08.08.99

## APPENDIX A

COPYRIGHT (C) 1990 GENERAL ELECTRIC COMPANY

```
task_table_index = INITIALIZATION_TASK
REPEAT
  WHILE (task_address <> SENTINEL_VALUE)
    AND (os_mode_change_status <> IMMEDIATE)
      task_page = page specified in task_table_buffer
      [task_table_index]
      task_address = address specified in task_table_buffer
      [task_table_index]
      IF task_address >= DATA_RAM_START
        perform change_data_ram_page(task_page)
      ELSE
        perform change_code_ram_page(task_page)
      ENDIF
      perform call to task_address
      increment task_table_index
    ENDWHILE
    previous_os_mode = current_os_mode
    current_os_mode = next_os_mode
    IF current_os_mode = previous_os_mode
      task_table_index = FIRST_REPEATED_TASK
    ELSE
      task_table_index = INITIALIZATION_TASK
      perform update_os_mode()
    ENDIF
  FOREVER
```



08.08.99

## APPENDIX B

COPYRIGHT (C) 1990 GENERAL ELECTRIC COMPANY

rom\_mode\_table:

|    |  |
|----|--|
| DB | PAGE_1                                       |
| DW | rom_initialization_task_table                |
| DB | PAGE_1                                       |
| DW | rom_demand_only_task_table                   |
| DB | PAGE_1                                       |
| DW | rom_demand_only_power_fail_task_table        |
| DB | PAGE_1                                       |
| DW | rom_demand_only_test_mode_task_table         |
| DB | PAGE_1                                       |
| DW | rom_optocom_mode_task_table                  |
| DB | PAGE_1                                       |
| DW | rom_std_protocol_mode_task_table             |
| DB | PAGE_1                                       |
| DW | rom_fail_safe_mode_task_table                |
| DB | PAGE_1                                       |
| DW | rom_fail_safe_test_mode_task_table           |
| DB | PAGE_1                                       |
| DW | rom_fail_safe_initialization_mode_task_table |
| DB | PAGE_1                                       |
| DW | rom_manufacturing_test_mode_task_table       |

08.08.99

## APPENDIX C

COPYRIGHT (C) 1990 GENERAL ELECTRIC COMPANY

rom\_task\_tables:

rom\_initialization\_task\_table:

|    |                                   |
|----|-----------------------------------|
| DB | PAGE_0                            |
| DW | initialization_mode_init          |
| DB | PAGE_0                            |
| DW | communication_task_initialization |
| DB | PAGE_0                            |
| DW | init_parameter_program            |
| DB | PAGE_0                            |
| DW | init_totals_task                  |
| DB | PAGE_0                            |
| DW | time_date_initialization          |
| DB | PAGE_0                            |
| DW | demand_calc_initialization        |
| DB | PAGE_0                            |
| DW | init_external_control_task        |
| DB | PAGE_0                            |
| DW | init_check_thresholds_task        |
| DB | PAGE_0                            |
| DW | init_reset_switch_control_task    |
| DB | PAGE_0                            |
| DW | display_task_init                 |
| DB | PAGE_0                            |
| DW | init_test_task                    |
| DB | PAGE_0                            |
| DW | self_test_high_level_init         |
| DB | PAGE_0                            |
| DW | high_level_initialization         |
| DB | OOE                               |
| DW | SENTINEL_VALUE                    |

rom\_demand\_only\_task\_table:

|    |                              |
|----|------------------------------|
| DB | PAGE_0                       |
| DW | demand_only_mode_init        |
| DB | PAGE_0                       |
| DW | allow_power_fail_interrupt   |
| DB | PAGE_0                       |
| DW | communication_monitor        |
| DB | PAGE_0                       |
| DW | prevent_power_fail_interrupt |
| DB | PAGE_0                       |

00.00.00

DW totals\_task  
DB PAGE\_0  
DW demand\_only\_time\_date\_task\_support  
DB PAGE\_0  
DW demand\_only\_demand\_calculation  
DB PAGE\_0  
DW allow\_power\_fail\_interrupt  
DB PAGE\_0  
DW prevent\_power\_fail\_interrupt  
DB PAGE\_0  
DW external\_control\_task  
DB PAGE\_0  
DW check\_thresholds\_task  
DB PAGE\_0  
DW reset\_switch\_control\_task  
DB PAGE\_0  
DW allow\_power\_fail\_interrupt  
DB PAGE\_0  
DW display\_task  
DB PAGE\_0  
DW inactive\_test\_mode  
DB PAGE\_0  
DW param\_pgm\_task  
DB PAGE\_0  
DW prevent\_power\_fail\_interrupt  
DB PAGE\_0  
DW self\_test\_task  
DB PAGE\_0  
DW allow\_power\_fail\_interrupt  
DB 00H  
DW SENTINEL\_VALUE

rom\_demand\_only\_power\_fail\_task\_table:

DB PAGE\_0  
DW demand\_only\_power\_fail\_mode\_init  
DB PAGE\_0  
DW demand\_only\_power\_fail\_totals  
DB PAGE\_0  
DW demand\_only\_time\_date\_task\_support  
DB PAGE\_0  
DW demand\_only\_demand\_calculation  
DB 00H  
DW SENTINEL\_VALUE

08.08.99

rom\_demand\_only\_test\_mode\_task\_table:

|    |                                    |
|----|------------------------------------|
| DB | PAGE_0                             |
| DW | test_mode_initialization_task      |
| DB | PAGE_0                             |
| DW | prevent_power_fail_interrupt       |
| DB | PAGE_0                             |
| DW | totals_test_support                |
| DB | PAGE_0                             |
| DW | demand_only_time_date_task_support |
| DB | PAGE_0                             |
| DW | allow_power_fail_interrupt         |
| DB | PAGE_0                             |
| DW | test_mode_demand_calculation       |
| DB | PAGE_0                             |
| DW | external_control_task              |
| DB | PAGE_0                             |
| DW | check_thresholds_task              |
| DB | PAGE_0                             |
| DW | reset_switch_control_task          |
| DB | PAGE_0                             |
| DW | display_task                       |
| DB | PAGE_0                             |
| DW | param_pgm_task                     |
| DB | PAGE_0                             |
| DW | active_test_mode                   |
| DB | 00H                                |
| DW | SENTINEL_VALUE                     |

rom\_optocom\_mode\_task\_table:

|    |                         |
|----|-------------------------|
| DB | PAGE_0                  |
| DW | optocom_mode_init       |
| DB | PAGE_0                  |
| DW | optocom_message_handler |
| DB | PAGE_0                  |
| DW | param_pgm_optocom       |
| DB | 00H                     |
| DW | SENTINEL_VALUE          |

rom\_std\_protocol\_mode\_task\_table:

|    |                        |
|----|------------------------|
| DB | PAGE_0                 |
| DW | std_protocol_mode_init |
| DB | PAGE_0                 |
| DW | std_protocol_task      |
| DB | PAGE_0                 |
| DW | param_pgm_optocom      |
| DB | 00H                    |
| DW | SENTINEL_VALUE         |

00:00:00

rom\_fail\_safe\_mode\_task\_table:

|    |                              |
|----|------------------------------|
| DB | PAGE_0                       |
| DW | fail_safe_mode_init          |
| DB | PAGE_0                       |
| DW | allow_power_fail_interrupt   |
| DB | PAGE_0                       |
| DW | communication_monitor        |
| DB | PAGE_0                       |
| DW | prevent_power_fail_interrupt |
| DB | PAGE_0                       |
| DW | fail_safe_totals             |
| DB | PAGE_0                       |
| DW | allow_power_fail_interrupt   |
| DB | PAGE_0                       |
| DW | display_task                 |
| DB | PAGE_0                       |
| DW | inactive_test_mode           |
| DB | PAGE_0                       |
| DW | param_pgm_task               |
| DB | PAGE_0                       |
| DW | prevent_power_fail_interrupt |
| DB | PAGE_0                       |
| DW | self_test_task               |
| DB | PAGE_0                       |
| DW | allow_power_fail_interrupt   |
| DB | 00H                          |
| DW | SENTINEL_VALUE               |

rom\_fail\_safe\_test\_mode\_task\_table:

|    |                               |
|----|-------------------------------|
| DB | PAGE_0                        |
| DW | test_mode_initialization_task |
| DB | PAGE_0                        |
| DW | allow_power_fail_interrupt    |
| DB | PAGE_0                        |
| DW | totals_test_support           |
| DB | PAGE_0                        |
| DW | test_mode_demand_calculation  |
| DB | PAGE_0                        |
| DW | check_thresholds_task         |
| DB | PAGE_0                        |
| DW | reset_switch_control_task     |
| DB | PAGE_0                        |
| DW | display_task                  |
| DB | PAGE_0                        |
| DW | param_pgm_task                |
| DB | PAGE_0                        |
| DW | active_test_mode              |
| DB | 00H                           |
| DW | SENTINEL_VALUE                |

08.08.99

rom\_fail\_safe\_initialization\_mode\_task\_table:

|    |                                    |
|----|------------------------------------|
| DB | PAGE_0                             |
| DW | fail_safe_initialization_mode_init |
| DB | PAGE_0                             |
| DW | communication_task_initialization  |
| DB | PAGE_0                             |
| DW | init_parameter_program             |
| DB | PAGE_0                             |
| DW | init_totals_task                   |
| DB | PAGE_0                             |
| DW | time_date_initialization           |
| DB | PAGE_0                             |
| DW | demand_calc_initialization         |
| DB | PAGE_0                             |
| DW | init_check_thresholds_task         |
| DB | PAGE_0                             |
| DW | init_reset_switch_control_task     |
| DB | PAGE_0                             |
| DW | display_task_init                  |
| DB | PAGE_0                             |
| DW | init_test_task                     |
| DB | PAGE_0                             |
| DW | self_test_fail_safe_init           |
| DB | PAGE_0                             |
| DW | high_level_initialization          |
| DB | 00H                                |
| DW | SENTINEL_VALUE                     |

rom\_manufacturing\_test\_mode\_task\_table:

|    |                              |
|----|------------------------------|
| DB | PAGE_0                       |
| DW | manufacturing_test_mode_init |
| DB | PAGE_0                       |
| DW | manufacturing_test_task      |
| DB | 00H                          |
| DW | SENTINEL_VALUE               |

08.08.99

# APPENDIX D

COPYRIGHT (C) 1990 GENERAL ELECTRIC COMPANY

ram\_mode\_table:

|    |  |              |
|----|--|--------------|
| DB | PAGE_1                                       |              |
| DW | tou_demand_initialization_task_table         |              |
| DB | PAGE_0                                       |              |
| DW | rom_demand_only_task_table                   |              |
| DB | PAGE_0                                       |              |
| DW | rom_demand_only_power_fail_task_table        |              |
| DB | PAGE_0                                       |              |
| DW | rom_demand_only_test_mode_task_table         |              |
| DB | PAGE_0                                       |              |
| DW | rom_optocom_mode_task_table                  |              |
| DB | PAGE_0                                       |              |
| DW | rom_std_protocol_mode_task_table             |              |
| DB | PAGE_0                                       |              |
| DW | rom_fail_safe_mode_task_table                |              |
| DB | PAGE_0                                       |              |
| DW | rom_fail_safe_test_mode_task_table           |              |
| DB | PAGE_0                                       |              |
| DW | rom_fail_safe_initialization_mode_task_table |              |
| DB | PAGE_0                                       |              |
| DW | rom_manufacturing_test_mode_task_table       |              |
| DB | PAGE_1                                       |              |
| DW | tou_demand_task_table                        |              |
| DB | PAGE_1                                       |              |
| DW | catch_up_mode_task_table                     |              |
| DB | PAGE_1                                       |              |
| DW | tou_demand_test_mode_task_table              |              |
| DB | PAGE_0                                       |              |
| DW | 0000H  | ;extra entry |
| DB | PAGE_0                                       |              |
| DW | 0000H  | ;extra entry |

08.08.99

APPENDIX E

COPYRIGHT (C) 1990 GENERAL ELECTRIC COMPANY

ram\_task\_tables:

tou\_demand\_initialization\_task\_table:

|    |                                   |
|----|-----------------------------------|
| DB | PAGE_0                            |
| DW | initialization_mode_init          |
| DB | PAGE_0                            |
| DW | high_level_initialization         |
| DB | PAGE_0                            |
| DW | communication_task_initialization |
| DB | PAGE_0                            |
| DW | init_parameter_program            |
| DB | PAGE_0                            |
| DW | init_totals_task                  |
| DB | PAGE_0                            |
| DW | time_date_initialization          |
| DB | PAGE_2                            |
| DW | init_prog_dates_task              |
| DB | PAGE_0                            |
| DW | demand_calc_initialization        |
| DB | PAGE_2                            |
| DW | init_season_change_task           |
| DB | PAGE_0                            |
| DW | init_external_control_task        |
| DB | PAGE_2                            |
| DW | init_rate_change_task             |
| DB | PAGE_2                            |
| DW | init_load_profile_task            |
| DB | PAGE_0                            |
| DW | init_check_thresholds_task        |
| DB | PAGE_0                            |
| DW | init_reset_switch_control_task    |
| DB | PAGE_0                            |
| DW | display_task_init                 |
| DB | PAGE_0                            |
| DW | init_test_task                    |
| DB | PAGE_0                            |
| DW | self_test_high_level_init         |
| DB | 00H                               |
| DW | SENTINEL_VALUE                    |



08.08.99

tou\_demand\_task\_table:

|    |                                   |
|----|-----------------------------------|
| DB | PAGE_2                            |
| DW | tou_demand_mode_init              |
| DB | PAGE_0                            |
| DW | abort_after_power_fail            |
| DB | PAGE_0                            |
| DW | communication_monitor             |
| DB | PAGE_0                            |
| DW | continue_after_power_fail         |
| DB | PAGE_0                            |
| DW | prevent_power_fail_interrupt      |
| DB | PAGE_0                            |
| DW | totals_task                       |
| DB | PAGE_0                            |
| DW | allow_power_fail_interrupt        |
| DB | PAGE_2                            |
| DW | tou_demand_time_date_task_support |
| DB | PAGE_2                            |
| DW | programmable_dates_task           |
| DB | PAGE_2                            |
| DW | tou_demand_demand_calculation     |
| DB | PAGE_2                            |
| DW | season_change_task                |
| DB | PAGE_2                            |
| DW | rate_change_task                  |
| DB | PAGE_2                            |
| DW | load_profile_recording_task       |
| DB | PAGE_0                            |
| DW | external_control_task             |
| DB | PAGE_0                            |
| DW | check_thresholds_task             |
| DB | PAGE_0                            |
| DW | reset_switch_control_task         |
| DB | PAGE_0                            |
| DW | abort_after_power_fail            |
| DB | PAGE_0                            |
| DW | display_task                      |
| DB | PAGE_0                            |
| DW | inactive_test_mode                |
| DB | PAGE_0                            |
| DW | param_pgm_task                    |
| DB | PAGE_0                            |
| DW | continue_after_power_fail         |
| DB | PAGE_0                            |
| DW | prevent_power_fail_interrupt      |
| DB | PAGE_0                            |
| DW | self_test_task                    |
| DB | PAGE_0                            |
| DW | allow_power_fail_interrupt        |
| DB | 00H                               |
| DW | SENTINEL_VALUE                    |

08.08.99

catch\_up\_mode\_task\_table:

|    |                                   |
|----|-----------------------------------|
| DB | PAGE_2                            |
| DW | catch_up_mode_init                |
| DB | PAGE_2                            |
| DW | prevent_power_fail_interrupt      |
| DB | PAGE_2                            |
| DW | totals_catch_up_support           |
| DB | PAGE_0                            |
| DW | allow_power_fail_interrupt        |
| DB | PAGE_2                            |
| DW | tou_demand_time_date_task_support |
| DB | PAGE_2                            |
| DW | programmable_dates_task           |
| DB | PAGE_2                            |
| DW | fast_catch_up_task                |
| DB | PAGE_2                            |
| DW | tou_demand_demand_calculation     |
| DB | PAGE_2                            |
| DW | season_change_task                |
| DB | PAGE_2                            |
| DW | rate_change_task                  |
| DB | PAGE_2                            |
| DW | load_profile_recording_task       |
| DB | 00H                               |
| DW | SENTINEL_VALUE                    |

08.08.99

tou\_demand\_test\_mode\_task\_table:

|    |                                   |
|----|-----------------------------------|
| DB | PAGE_0                            |
| DW | test_mode_initialization_task     |
| DB | PAGE_0                            |
| DW | prevent_power_fail_interrupt      |
| DB | PAGE_0                            |
| DW | totals_test_support               |
| DB | PAGE_2                            |
| DW | tou_demand_time_date_task_support |
| DB | PAGE_2                            |
| DW | programmable_dates_task           |
| DB | PAGE_0                            |
| DW | test_mode_demand_calculation      |
| DB | PAGE_2                            |
| DW | season_change_task                |
| DB | PAGE_2                            |
| DW | rate_change_task                  |
| DB | PAGE_2                            |
| DW | load_profile_recording_task       |
| DB | PAGE_0                            |
| DW | external_control_task             |
| DB | PAGE_0                            |
| DW | check_thresholds_task             |
| DB | PAGE_0                            |
| DW | reset_switch_control_task         |
| DB | PAGE_0                            |
| DW | allow_power_fail_interrupt        |
| DB | PAGE_0                            |
| DW | display_task                      |
| DB | PAGE_0                            |
| DW | param_pgma_task                   |
| DB | PAGE_0                            |
| DW | active_test_mode                  |
| DB | 00H                               |
| DW | SENTINEL_VALUE                    |

08.08.99

P 692 28 850.3-08

Patentansprüche:

- 5 1. Programmierbare elektronische Festkörper-Registerkomponente (150) für ein elektrisches Energiemeßgerät, enthaltend:  
eine Modus-Tabelle (300);  
eine Anzahl von Aufgaben-Tabellen (302,304);  
eine Anzahl von Moduln;
- 10 wobei die Modus-Tabelle ein Array von Aufgaben-Zeigern auf entsprechende der Anzahl von Aufgaben-Tabellen aufweist;  
wobei jede der Anzahl von Aufgaben-Tabellen ein Array von Modul-Zeigern auf entsprechende der Anzahl von Moduln aufweist;
- 15 wobei jede der Anzahl von Moduln eine Anzahl von Befehlen aufweist, die während der Abarbeitung von einer entsprechenden Aufgabe auszuführen sind, die in wenigstens einer der vorbestimmten Gruppen von Aufgaben enthalten ist, und  
wobei jede der vorbestimmten Gruppen von Aufgaben durch
- 20 eine der Aufgaben-Tabellen definiert ist und einem vorbestimmten Betriebsmodus entspricht, wie beispielsweise einem Verwendungszeit- und Bedarfs-Betriebsmodus oder einem Nur-Bedarfs-Betriebsmodus; und  
eine Mikroprozessor-Steuereinrichtung (152) zum Betreiben der Registerkomponente in dem vorbestimmten Betriebsmodus,
- 25 indem eine entsprechende vorbestimmte Gruppe von Aufgaben ausgeführt wird, wobei die Mikroprozessor-Steuereinrichtung gekennzeichnet ist durch:  
eine Einrichtung zum Indexieren (204,206) auf einen Aufgaben-Zeiger in der Modus-Tabelle, um dadurch eine der vorbestimmten Gruppen von Aufgaben zu wählen, wobei der Aufgaben-Zeiger einer gewählten Aufgaben-Tabelle entspricht,
- 30 eine Einrichtung zum Ausführen (208,220-230,234) der einen der vorbestimmten Gruppen von Aufgaben, indem wiederholt die gewählte Aufgaben-Tabelle auf entsprechende Modul-Zeiger indexiert wird, und
- 35 eine Einrichtung zum Ausführen (232) der Anzahl von Moduln entsprechend den entsprechenden Modul-Zeigern.

08.08.99

- 5 2. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 1, wobei die Modus-Tabelle, die Anzahl von Aufgaben-Tabellen und die Anzahl von Moduln in einem Speicher (164) enthalten sind, der einen Festwertspeicher und einen Arbeitsspeicher aufweist, und wobei die Einrichtung zum Ausführen von einer der vorbestimmten Gruppen von Aufgaben eine Einrichtung aufweist zum Ausführen der einen der vorbestimmten Gruppen von Aufgaben in dem Arbeitsspeicher.
- 10 3. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 2, wobei ferner eine Eingabe/Ausgabe-Einrichtung (156), die auf eine Betätigung durch einen Benutzer zum Wählen des programmierten Betriebsmodus anspricht, und eine Einrichtung zum Runterladen (154) der Aufgaben-Tabellen entsprechend dem programmierten Betriebsmodus in den Arbeitsspeicher vorgesehen sind.
- 15 4. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 3, wobei die Einrichtung zum Runterladen der Aufgaben-Tabellen eine Einrichtung zum Runterladen (154) der Aufgaben-Tabellen entsprechend dem programmierten Betriebsmodus von dem Festwertspeicher in den Arbeitsspeicher aufweist.
- 20 5. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 4, wobei die Eingabe/Ausgabe-Einrichtung ein Flüssigkristalldisplay aufweist.
- 25 6. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 1, wobei die Arrays von Modul-Zeigern für jede der Aufgaben-Tabellen einen Modus-Initialisierungszeiger (304, Eintrag = 0) entsprechend einem entsprechenden Modus-Initialisierungsmodus enthält, und wobei die Einrichtung zum Ausführen der Anzahl von Moduln eine Einrichtung aufweist zum Ausführen des entsprechenden Modus-Initialisierungsmoduls nur einmal während des programmierten Betriebsmodus.
- 30 35

03.05.99

- 5 7. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 3, wobei die Arrays von Modul-Zeigern für jede der Aufgaben-Tabellen einen Modus-Initialisierungszeiger (304, Eintrag = 0) entsprechend einem entsprechenden Modus-Initialisierungsmodul aufweist, und wobei die Einrichtung zum Ausführen der Anzahl von Modulen eine Einrichtung zum Ausführen (218) des entsprechenden Modus-Initialisierungsmoduls nur einmal während des programmierten Betriebsmodus aufweist.
- 10 8. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 7, ferner enthaltend:  
eine Einrichtung zum Halten (220) der Einrichtung zum Ausführen der einen der vorbestimmten Gruppen von Aufgaben vor dem Indexieren der gesamten gewählten Aufgaben-Tabelle; und  
15 eine Einrichtung zum Initialisieren (216) der Einrichtung zum Runterladen der Aufgaben-Tabellen entsprechend einem anderen programmierten Betriebsmodus von dem Festwertspeicher in den Arbeitsspeicher.
- 20 9. Programmierbare elektronische Festkörper-Registerkomponente nach Anspruch 2, wobei der Speicher und die Mikroprozessor-Steuereinrichtung in einer Anwendungs-spezifischen integrierten Schaltung enthalten sind.

25

0534 583

08.08.88

1/4

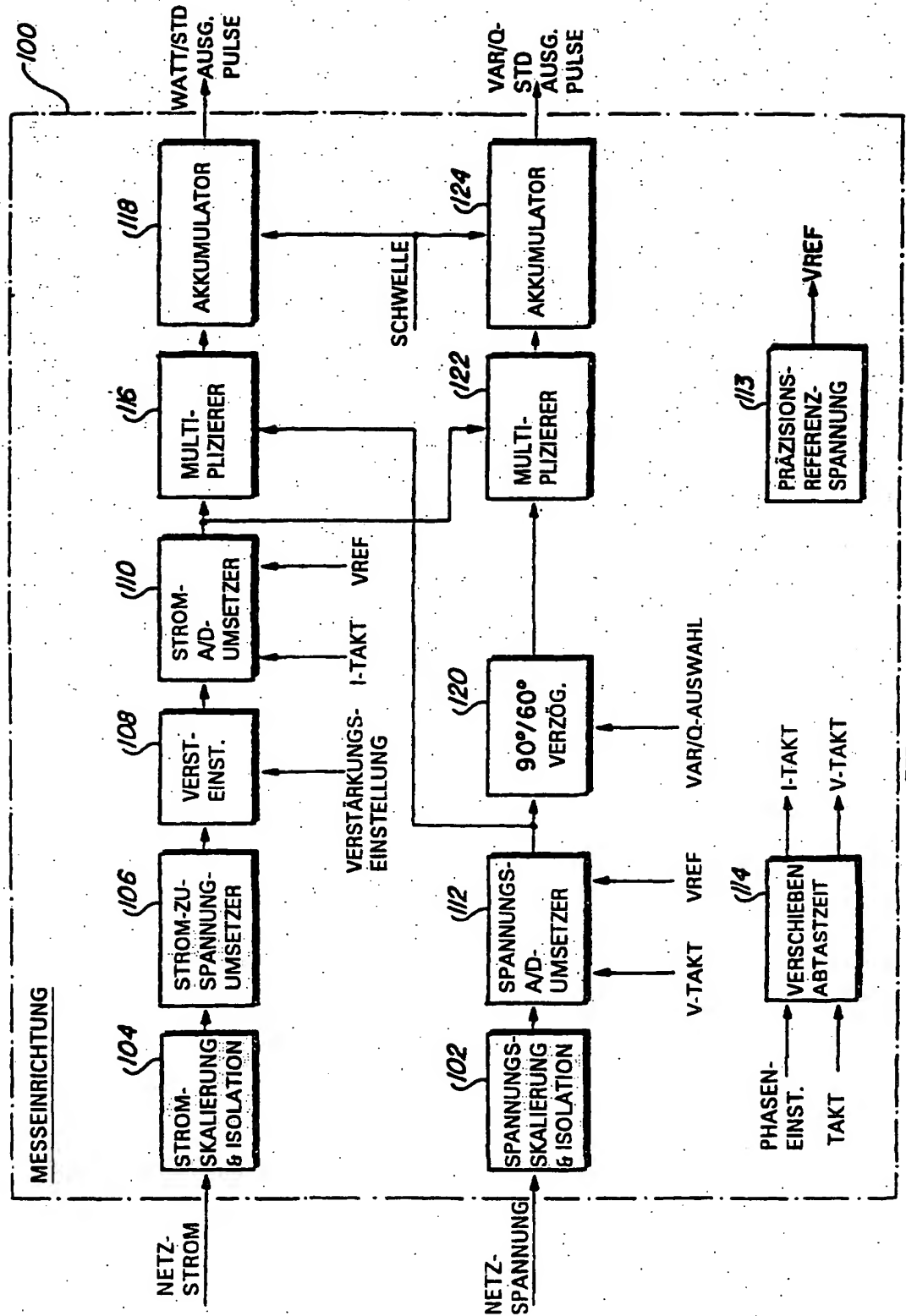


Fig. 1

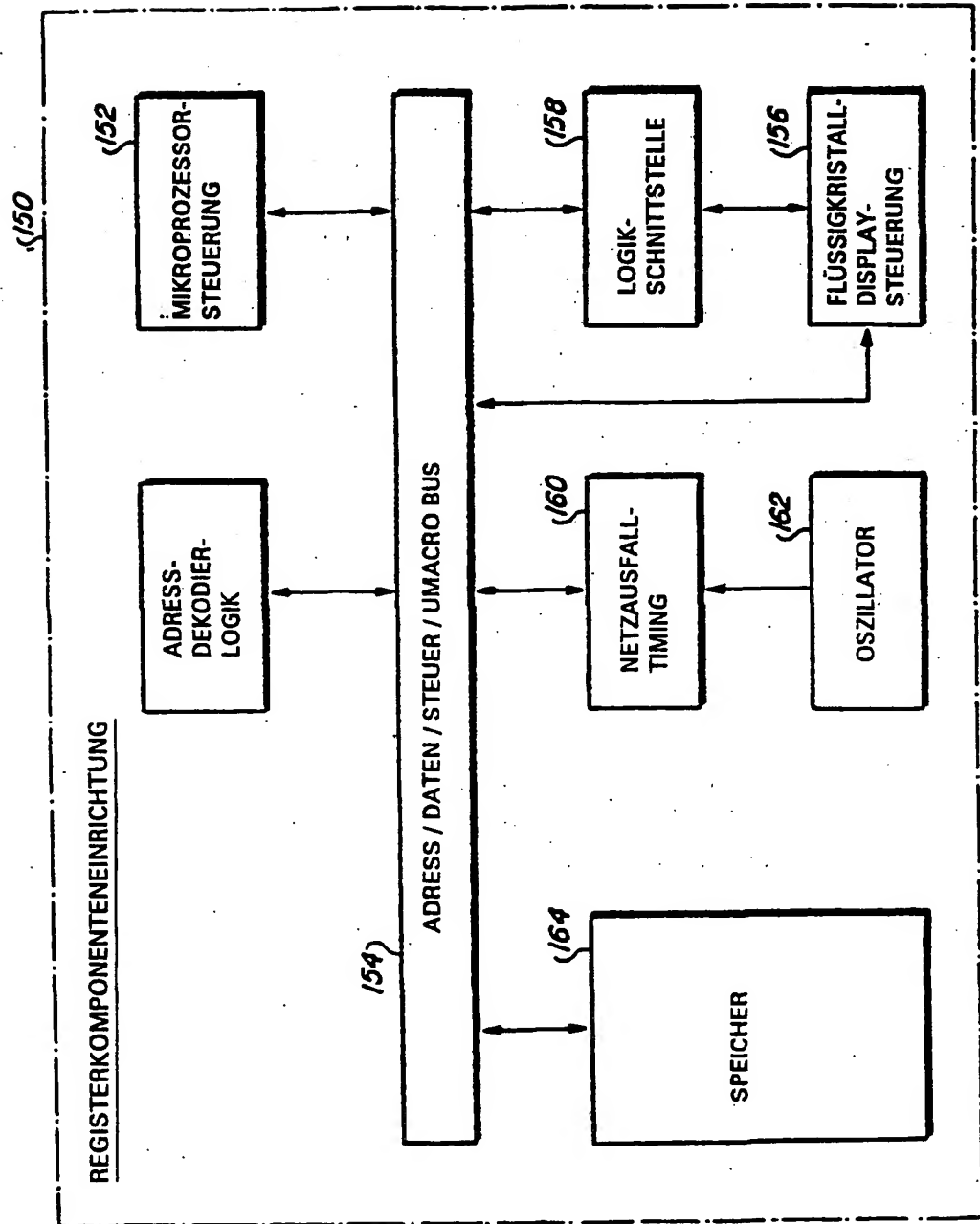


Fig. 2



08.06.99

3/4

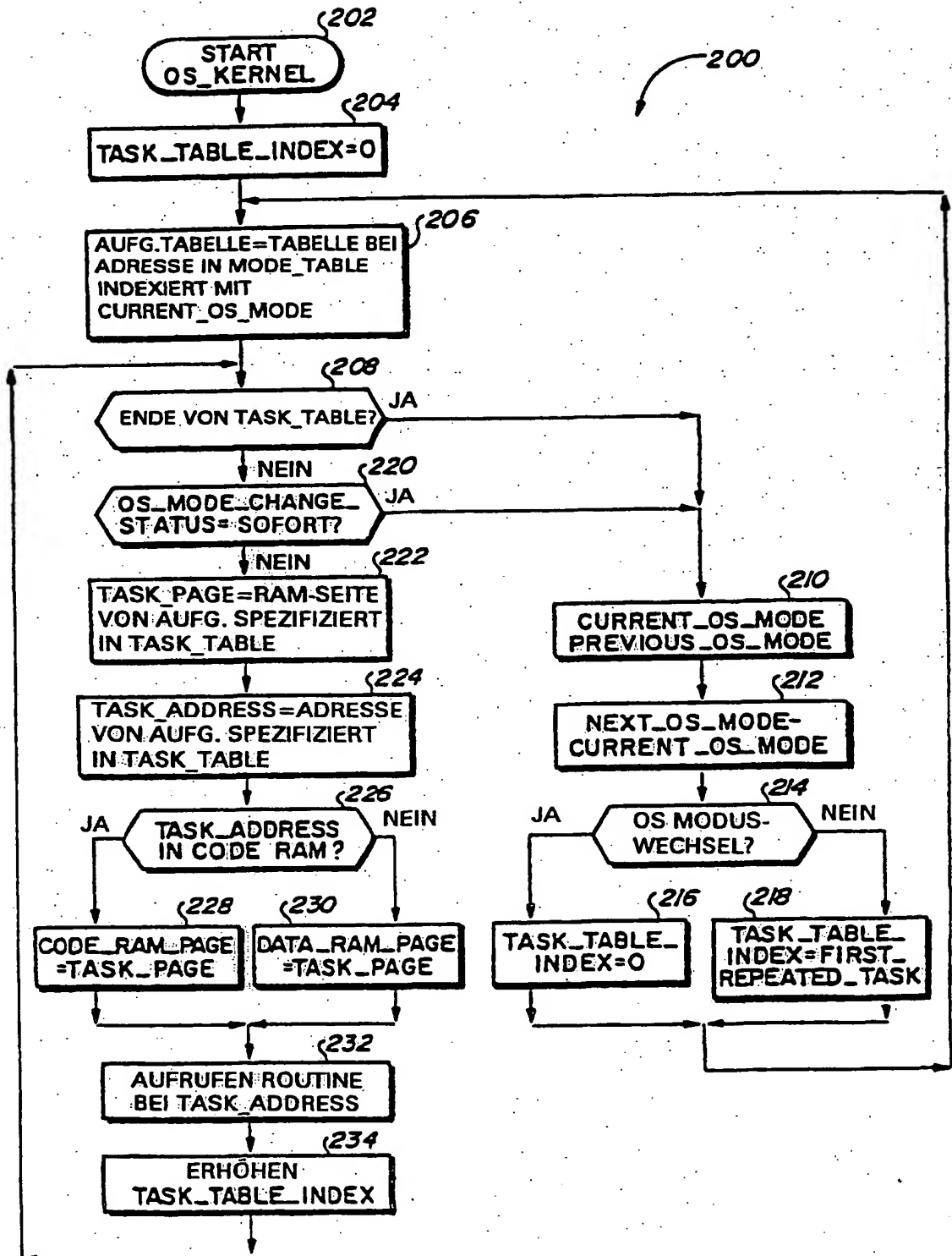


FIG. 3

08.08.99

4/4

FIG. 4

| TABELLE | EINTR | BYTE | WERT                                   |
|---------|-------|------|--|
| TASKTBL | 0     | 0    | SPEICHERSEITE FÜR MODE-INITIAL.AUFGABE |
|         | 0     | 1    | ADRESSE FÜR MODE-INITIAL.AUFGABE       |
|         | 1     | 3    | SPEICHERSEITE FÜR AUFGABE 1            |
|         | 1     | 4    | ADRESSE FÜR AUFGABE 1                  |
|         |       |      |  |
|         |       |      |  |
|         |       |      |  |
|         | n     | 3n   | SPEICHERSEITE FÜR AUFGABE n            |
|         | n     | 3n+1 | ADRESSE FÜR AUFGABE n                  |
|         | n+1   | 3n+3 | UNBENUTZT                              |
|         | n+1   | 3n+4 | MARKIERUNGSWERT (0)                    |

300

| TABELLE | EINTR | BYTE | WERT                             |
|---------|-------|------|----------------------------------|
| MODETEL | 0     | 0    | SPEICHERSEITE FÜR AUFG.TABELLE 1 |
|         | 0     | 1    | ADRESSE FÜR AUFG.TABELLE 1       |
|         | 1     | 3    | SPEICHERSEITE FÜR AUFG.TABELLE 2 |
|         | 1     | 4    | ADRESSE FÜR AUFG.TABELLE 2       |
|         |       |      |                                  |
|         |       |      |                                  |
|         |       |      |                                  |
|         | i     | 3i   | SPEICHERSEITE FÜR AUFG.TABELLE i |
|         | i     | 3i+1 | ADRESSE FÜR AUFG.TABELLE i       |
|         | i+1   | 3i+3 | UNBENUTZT                        |
|         | i+1   | 3i+4 | MARKIERUNGSWERT (0)              |

302

304

| TABELLE | EINTR | BYTE | WERT                                   |
|---------|-------|------|--|
| TASKTBL | 0     | 0    | SPEICHERSEITE FÜR MODE-INITIAL.AUFGABE |
|         | 0     | 1    | ADRESSE FÜR MODE-INITIAL.AUFGABE       |
|         | 1     | 3    | SPEICHERSEITE FÜR AUFGABE 1            |
|         | 1     | 4    | ADRESSE FÜR AUFGABE 1                  |
|         |       |      |  |
|         |       |      |  |
|         |       |      |  |
|         | n     | 3n   | SPEICHERSEITE FÜR AUFGABE n            |
|         | n     | 3n+1 | ADRESSE FÜR AUFGABE n                  |
|         | n+1   | 3n+3 | UNBENUTZT                              |
|         | n+1   | 3n+4 | MARKIERUNGSWERT (0)                    |